Science-Return Modeling and Simulation[1]

Meemong Lee, Richard J. Weidner, Shin-Ywan Wang, Brian M. Lininger
Jet Propulsion Laboratory
4800 Oak Grove, Pasadena, CA 91109
818-354-2228
meemong.lee@jpl.nasa.gov,
richard.weidner@jpl.nasa.gov,
shin-ywan.wang@jpl.nasa.gov,
brian.m.lininger@jpl.nasa.gov

Abstract—Mission system designers formulate subsystem performance requirements based on the science-return objectives of a mission. The overall system architecture is optimized through inter-subsystem performance trade-off analysis. Design optimization with respect to the entire mission lifecycle is also greatly desired in order to reduce the overall mission development cycle and operational risk. To facilitate the lifecycle-wide design validation process, the Mission Simulation and Instrument Modeling (MSIM) group at the Jet Propulsion Laboratory has been developing mission operation modeling methods and operation-behavior simulation system architectures. The modeling and simulation efforts have been applied to science-return probability analysis as well as to science scenario optimization. In this paper, an on-going effort to integrate science-return modeling and simulation with JPL's institutional mission system design process will be presented. The integration will enable science-return validation at each stage of the design process. Science-return modeling and simulation will be discussed with respect to science scenario generation, science-centric mission system representation, and operation-behavior synthesis and analysis.

TABLE OF CONTENTS

1. INTRODUCTION

The term "science-return" in this paper refers to the science data products planned to be acquired during a deep space mission. A complex set of operations involving multiple subsystems is required to ensure the desired quantity and quality of each science data product. The complexity of the science operation is tightly coupled with the mission subsystem architecture in terms of the performance and the resource usage profile of each subsystem. The integrated system performance must be able to achieve the desired data quality with respect to sample resolution and signal fidelity. The on-board resources, such as processor, memory, bus bandwidth, must be sufficient to acquire and to process the desired data volume. Also, the resource allocation must be precisely coordinated throughout the entire dataflow stages, from the sensing of scientific phenomena to the reception of the telemetry on the ground.

The Science-Return Modeling and Simulation (SRMS) task at JPL develops modeling and simulation systems for evaluating integrated system performance, profiling resource usage, and planning inter-subsystem operation coordination. In order to ensure the science-return, the SRMS addresses a systematic tracking of the mission system properties during the entire mission lifecycle as well as a phase-specific impact analysis. The main emphasis of the current development is to streamline the early design phase of the Project Design Center (PDC) at JPL with the science-return validation process.

The SRMS implements an end-to-end process that integrates three major aspects of the project design—science objective, mission system architecture, and mission design. The SRMS activity groups the mission applications into three types—fly-by, orbit, and in-situ, depending on the relationship between the spacecraft and the science target. Each mission type provides a set of type-specific assumptions so that the description of the science objective, mission system architecture, and mission design can be simplified. The end-to-end process is implemented in a concurrent engineering environment so that the design products from multiple design activities can be incrementally integrated. Figure 1 illustrates the iteration flow and the integration process of the three design activities.

The high-level science scenario generation system provides an easier and more flexible syntax for specifying the observation activities. It also provides

[1]

progressive specification methods to translate the objective into operational activities as the system design matures. The science scenario generation process involves a science objective description language, model-based analysis for command, science target, and operation duration, and an activity sequence generator. The technical details of the high-level science scenario generation system will be discussed in Section 2.

The spacecraft property knowledge system provides a science-centric design parameter database and a subsystem virtual prototyping based on the performance parameters. The design parameters that are closely related to the science observation are extracted from the institutional mission system design database to form the science-centric design parameter database. The science-return relevant properties refer to the design parameters that affect signal quality, acquisition, processing, and transmission, of the science data. Section 3 describes the current implementation of the spacecraft property knowledge system with respect to its content and usage.

The virtual operation system simulates the operation behavior of the mission system with respect to resource usage, data product generation, and geometric alignment of the spacecraft. The operation behavior simulation presents the time and space relationship of the observation dynamics, the observed phenomena, data product synthesis, and data processing and transmission. In Section 4, agent-based virtual spacecraft integration, system-level operation behavior synthesis, and comprehensive monitoring of the science data product acquisition and delivery process will be discussed.

2. Science Scenario Generation

One of the most important steps in science-return validation is to capture the science objective of a mission accurately in terms of the relationship between the observer and the target phenomena to be observed [1]. The accurate capturing must allow a progressive specification as the knowledge of the observer (i.e., spacecraft and instrument) and the target phenomena increases. The progressive specification requires an abstract representation for describing the desired observation conditions and a set of condition analysis mechanisms [2]. The abstract representation is used to define phenomena properties, observation conditions, resource availabilities, and various observation related situations. The condition analysis mechanisms translate each condition into a time range when the condition is met utilizing the knowledge of the spacecraft properties [3,4].

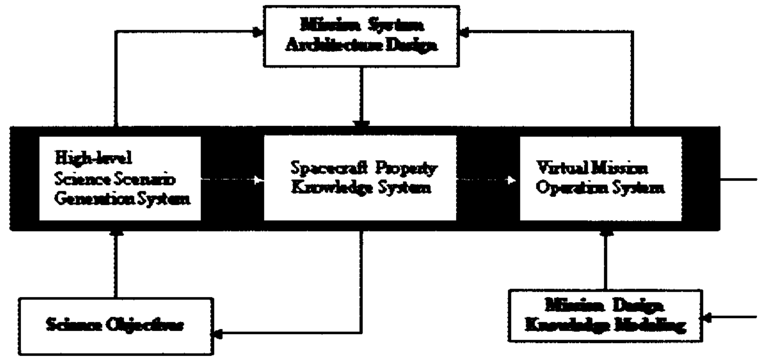


Figure 1 Progressive End-to-End Process

The science scenario generation system of the SRMS consists of three software modules, a high-level observation scenario language, a model-based condition analyzer, and a sequence composer. The high-level observation scenario language addresses the abstract representation syntax that enables scientists to express their observation activities in an event-driven manner. The model-based condition analyzer interacts with the spacecraft knowledge system to translate the event conditions into a specific time range. Finally, the sequence composer aligns and optimizes the timing of the observation activities and generates a set of time-stamped commands for involved subsystems.

The observation scenario syntax is composed of a list of observation activities. Each observation activity consists of an initial condition, a target, and a series of observation events for all subsystems involved in the observation. An observation scenario may be composed of multiple activities. The order of activities is determined based on optimal resource criteria (e.g., total duration, storage usage, down-link time, etc.) of performing all of the activities. The spacing between activities indicates the time involved in turning the spacecraft system from one target position to the next. Figure 2 illustrates the user interface of the science scenario generation system for defining the observation activities.

The target of a given observation activity may be described as a specific target name or as a target type with a desired target property. The available target types are Star, Sky, Planet, and Encounter. The target properties are Near/Nearest, Bright/Brightest, Dark/Darkest, etc. The degree of Near, Bright, and Dark can be defined in the scenario. The target types and target properties are provided so that an observation scenario can be composed in an abstract manner. A different target may be chosen for the same scenario depending on the spacecraft system state and the requested observation time range. Figure 3 illustrates the user interface for defining a set of target types. For example, the target property could be a patch of a dark sky where the dark sky is defined to be an area with less than 5 stars of visible magnitude greater than 10.

The subsystem event is described with an event condition and subsystem operation commands. The event condition is defined as a logical combination of three types of conditions—target condition, time condition, and command condition. The target condition is used to express the necessary target state during the event operation. The supported target states include distance, apparent size, phase angle, etc. The time condition is used to express the required time between events within a subsystem. The command condition is used to express interdependency and concurrency of the events among the multiple subsystems. In order to resolve the event conditions of each activity, target condition, time condition, and command condition, a set of condition analysis software modules are employed. The condition analysis software modules estimate the range of time when the event conditions can be satisfied by interfacing with the subsystem property knowledge system and the mission design knowledge system. Currently, three types of the condition analysis software modules have been developed, a science target analyzer, an operation duration analyzer, and a resource analyzer.

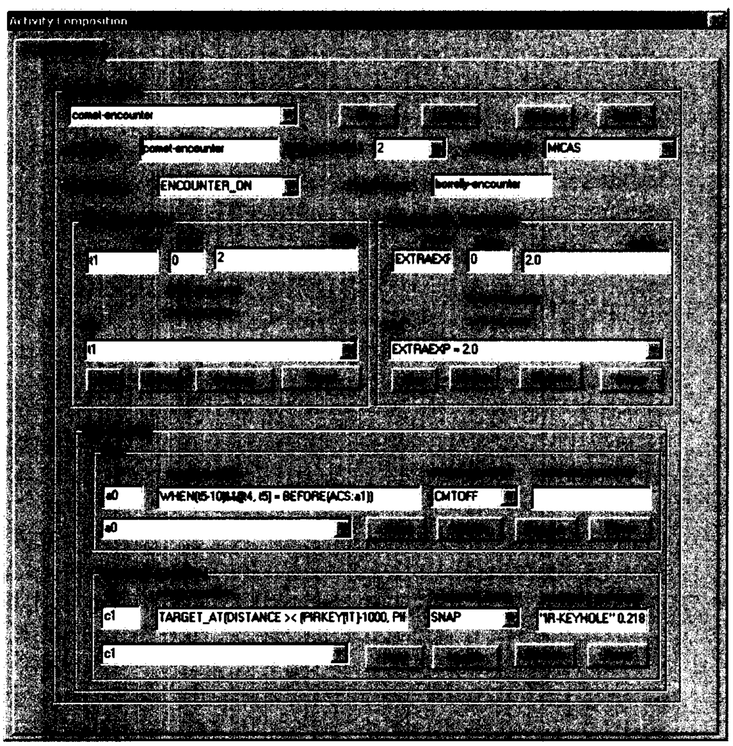


Figure 2 User Interface for Activity Definition

During the operation sequence composition involving multiple observation activities, an additional sequence that allows transition from one observation activity to the next is composed automatically. During the transition sequence composition, the required resource (e.g., turn duration) is also analyzed in order to optimize the order of the observation activities. The science target analyzer selects candidate targets and evaluates each candidate target for the desired condition and verifies the optimality with respect to other mission constraints. The operation duration analyzer evaluates the difference between the current state and the desired state and requests the corresponding subsystem to resolve the difference. Finally, the resource analyzer predicts the time range when the required resource may be available as well as provide the current level of available resources. The resource analyzer will be discussed in more detail in the next section in relation to the power and the telecom subsystem property modeling.

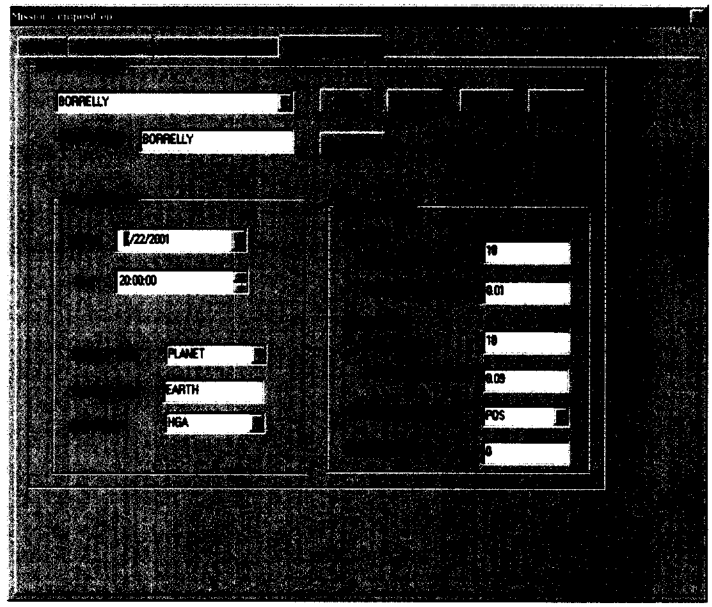


Figure 3 User Interface for Target Definition

3. MISSION SYSTEM PROPERTY REPRESENTATION

The SRMS represents a mission system in alignment with the mission system architecture defined by the Project Design Center (PDC) at JPL. The mission system architecture organizes the spacecraft system with a set of subsystems including navigation, attitude and articulation control, command and data handling (C&DH), power, telecom, and instrument subsystems. Also, the system level operation design activities are performed under mission, science, and ground [5]. The design parameters that are closely related to the science observation are extracted from the PDC design database to form the science-centric design parameter database. The science observation relevant properties refer to the design parameters that affect signal quality, acquisition, processing, and transmission, of the science data. The properties can include sensor performance, navigation accuracy, pointing control accuracy, on-board processing, on-board storage, antenna size, and so on.

The system properties of the above subsystems are categorized into three inter-operational model types—structure, performance, and operation. The structure model type represents the geometric shape of the spacecraft system as well as subsystems including their articulation mechanisms. The performance model type represents the capability ranges of the subsystem functions. Finally, the operation model type represents the control modes, resource usage profiles, and operational constraints. One of the challenges of the property modeling process described above is decomposition of the tightly coupled system properties into the above three model types. For example, the downlink performance of an antenna depends on the range, the elevation angle with respect to the receiving DSN antenna, and the encoding method. In order to decouple the operation conditions from the performance properties, the performance modeling of the telecom subsystem requires interface to an off-line analysis process to compose a downlink rate table for angles as shown in figure 4.

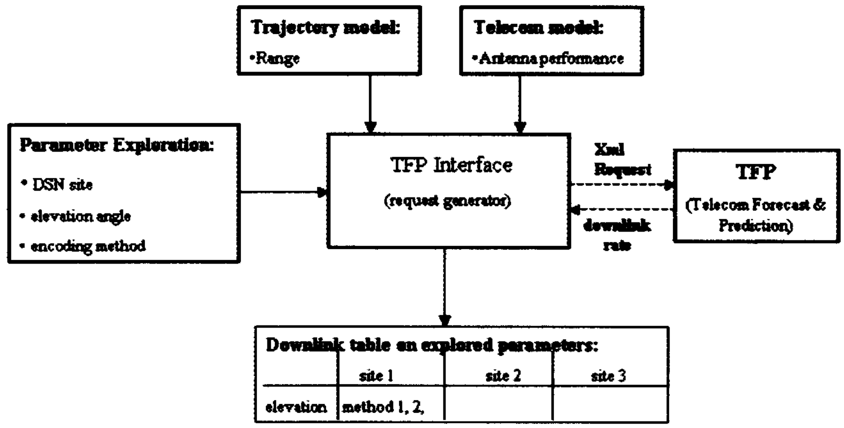


Figure 4 TFP Analysis Interface Protocol

After the downlink table has been generated, the telecom performance model is composed as an algorithm that utilizes the table for downlink rate estimation during the downlink operation simulation. As shown in Figure 5, the operation scenario simulation provides the encoding method, range, DSN site, and elevation angle to the downlink rate estimator. The operation-specific downlink performance table generated from this process is utilized for operation scenario optimization, operation cost analysis, and mission design. In order to support the two-step approach to formulate the performance model of a subsystem, the SRMS implements a subsystem-generic interface protocol that can interact with a wide range of remote analysis tools as discussed below.

As mentioned earlier, the off-line downlink rate analysis process is supported by the Telecom Forecast and Prediction system (TFP) developed by the Telecom Operation Group at JPL. The TFP interface module is implemented by employing the remote tool interface protocol which consists of the request rule script, rule-based parameter exploration, and request generation. Both the request rule and the request are composed in XML. The request rule script defines a list of parameters and inter-parameter dependencies. For each parameter, available values, their ranges, and default settings are specified. The inter-parameter dependency is described with an additional array of availability flags that are defined as a function of dependent parameter settings.

The TFP interface module accepts three types of information to compose a request to the TFP- range, antenna model, and a list of parameters to be explored. For each range specified, the TFP interface module reviews the options for each parameter to be explored utilizing a pre-defined parameter rule file. The TFP interface allows two types of exploration methods, Single and All. The Single method allows only one parameter to be explored at a time, while the All method allows exploration of the entire combination of all possible values.

In case of the All method, parameters are grouped into independent parameters and dependent parameters in order to reduce the search space. The dependent parameters are arranged in a tree structure and the valid values are traversed from the root of the tree. Based on the dependency chart, a tree is constructed for the number of parameters involved. The number of valid values in the first parameter determines the number of trees to be constructed. As the value is set for the first parameter, the values of the second parameter can be determined and the process repeats until it reaches the last parameter in the tree. The validated data set created from the above process is then combined with the value combination of the dependent parameters.

As the exploration process is performed, an XML request is generated with the valid parameter values and it is sent to the web-based request server. The web-based request server generates a request script and executes the TFP analysis program. The analyzed downlink rate result is sent back to the TFP interface via the web-based request server. The interface protocol for composing the downlink performance involves an XML-based request rule specification, a rule-based exploration mechanism, and an automated XML-based request generation. The interface protocol can be easily adapted to other performance analysis tools.

After the properties are modeled for the subsystems, virtual prototypes of the subsystems are developed. The subsystem virtual prototype is defined to be a software implementation of the subsystem properties that can be employed to perform the functions of the subsystem. The functions of a subsystem that the SRMS is interested in include operation command executions and diagnostics of the execution process via observable states. The SRMS currently provides virtual prototypes for six subsystems—navigation, attitude control, instrument, telecom, power, and C&DH. The virtual prototyping is performed based on the properties of the six subsystems that are specified in the spacecraft property knowledge system. The six virtual prototypes are integrated to construct a virtual spacecraft system that can generate integrated system states while performing science observation activities. The virtual spacecraft system will be discussed in the following section with respect to its integration architecture and operation simulation.

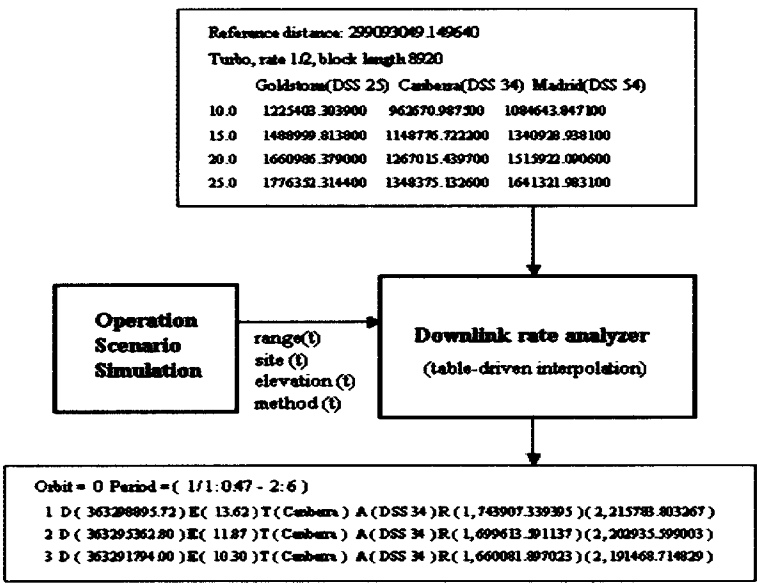


Figure 5 Operation-Level Downlink Rate Analysis

The term "prototyping" is used to emphasize the idea that the virtual subsystem can be treated as if it were a physical prototype subsystem with respect to command and data handling, mission data product generation, and resource usage behavior. Each virtual prototype is composed of three sub modules, a command handler, a property information server, and a time-based state generator. Depending on the complexity of the sub module, it can be implemented as either a subroutine or a remote process.

The command handler of a subsystem simulates the function of the software module of the subsystem in terms of receiving and verifying the incoming commands, planning the execution of the verified commands, interacting with the hardware devices, and providing the data/information to the external world. The property information server interfaces with the subsystem model providing a model script parsing, property model class construction, and derived information analysis. The time-based state generator propagates execution of a command after the command handler initiates the command. The propagation of a command execution indicates updating of the subsystem state with respect to the external world as well as internal resources as the command execution progresses.

4. Operation Behavior Analysis

The operation behavior analysis may be performed in two levels, the subsystem level and the system level. The subsystem level operation behavior analysis models the performance variation with respect to the operation conditions. The downlink rate modeling of the telecom subsystem described in the previous section is an example of the subsystem-level operation behavior analysis. The system-level operation behavior analysis evaluates the probability of achieving the ultimate science-return with respect to the mission design and mission system properties. The SRMS implements a virtual spacecraft that generates telemetry records, Micro-Helm, a 3×2 PC array, that supports real-time telemetry visualization by integrating the mission environment models and subsystem states in multiple perspectives, and a set of subsystem state analyzers that evaluate subsystem-specific performance defects.

In order to perform a desired mission operation as an integrated spacecraft system, the SRMS integrates the six subsystem virtual prototypes to construct a virtual spacecraft that can be commanded for science observation. The integration of the six virtual subsystems involves three levels of coordination. First, the operation scenario level for on-board command sequence planning and subsystem commanding. Second, the command execution level for time-based subsystem state integration and representation. Finally, the target phenomena integration level for instrument operation control and science data product generation. These coordination levels involve multiple disciplines, multiple processes, and multiple computational platforms. The SRMS utilizes intelligent mission model agents that are composed of domain-intelligent, platform-independent, and network-friendly information exchange components.

The intelligent mission model agent is capable of understanding and interacting with its environment on the behalf of its user, of moving about the web, and of forming and executing rudimentary decisions [6]. The information agent delivers useful information to the user by actively performing search, access, and retrieval of relevant information. The simulation of the mission design process requires intelligent information agents that search, access, and retrieve design information including spacecraft trajectory, planet ephemerides, planet kinematics, etc. The combination of datasets from multiple sources is often required to form particular information. Polymorphic functionality is required that can recognize different forms and transform them into a desired product. The technical details of the intelligent mission model agents are presented in the paper "Component-based Implementation of Agents and Brokers for Design Coordination" [5].

The operation behavior simulation generates two types of operation states, the true operation state and the estimated operation state. The real operation state indicates the actual spacecraft state with respect to the environment and the science targets while the estimated operation state indicates the state realized by the spacecraft system [7]. The real operation state can be monitored via Micro-Helm during the simulation. The estimated operation state is recorded as telemetry data as in the real mission case. The science data products are included in the telemetry data. The difference between the real operation state and estimated operation state is simulated based on the knowledge error specification of each subsystem. By introducing the predicted-knowledge-error range, the operation behavior simulation can illustrate the impact of the knowledge error on the operation planning. Also, the resulting science data products can comprehensively inform the impact on the science-return.

The science product generation [8,9] is the ultimate validation of the mission system, and it is one of the most challenging processes in the virtual mission project. Science product generation involves high-fidelity models of a target system, instrument system, and spacecraft system, and requires extensive computation. For example, simulation of an image acquired during the observation of an extended target involves per-pixel ray tracing of the reflected sunlight where the tracing geometry changes as the position and attitude of the spacecraft changes during the exposure duration. The spectral signature of the reflected sunlight also changes depending on the surface material and the spectral sensitivity profile of the instrument. The integration of the complex geometric, dynamic and radiometric relationships among the Sun, targets, spacecraft system, and instrument provide the operability validation that cannot be acquired with science product generation processes alone.

It is important to note that high-resolution data in the simulation indicates the level of detail in the information, not the accuracy of information. During the design phase, a detailed description may not be made with a high level of certainty. However, if the uncertainty range can be specified, the high-resolution measurements can be simulated within the specified uncertainty range, thus providing predicted impacts of the system design on the ultimate science return. Figures 6 and 7 illustrate synthetic data product examples during a fly-by mission and an in-situ mission respectively.

The virtual telemetry data products are organized following the real telemetry data products with respect to the data structure and sampling frequency. The Micro-Helm implements a time-synchronized operation behavior monitoring system involving six visualization clients. The six clients visualize instrument data acquisition, spacecraft attitude and articulation (solar panel and antenna), DSN connection, subsystem attitude geometry, power charge/load profile, and ground data system. Each visualization client runs on a window-based PC platform and a telemetry record server distributes the telemetry information to the six clients while synchronizing them at regular intervals. Each visualization client is equipped with a set of models that can be integrated with the telemetry data so that the telemetry data can be interpreted in a physically meaningful manner.

5. SUMMARY

The above three sections presented the end-to-end process of the SRMS developed for science-return validation during the early design phase. The end-to-end process integrates three major design activities-- science observation, mission system architecture, and mission operation. The integration involves three software systems, a high-level science scenario generation system, spacecraft property knowledge system, and virtual mission operation system. The technical objectives of each software system were discussed with respect to streamlining of the information flow, modeling and simulation of the properties, and progressive engineering of the design process.

The spacecraft property knowledge system of the SRMS defines three model types to capture the property of a subsystem, a performance model, an operation model, and a structure model. An interface protocol to a high-fidelity analysis tool was described with respect to request rule interpretation, request composition, and parameter option exploration. The SRMS currently interacts with two high-fidelity performance analysis tools, TFP and MMPAT (Multi-Mission Power Analysis Tool).

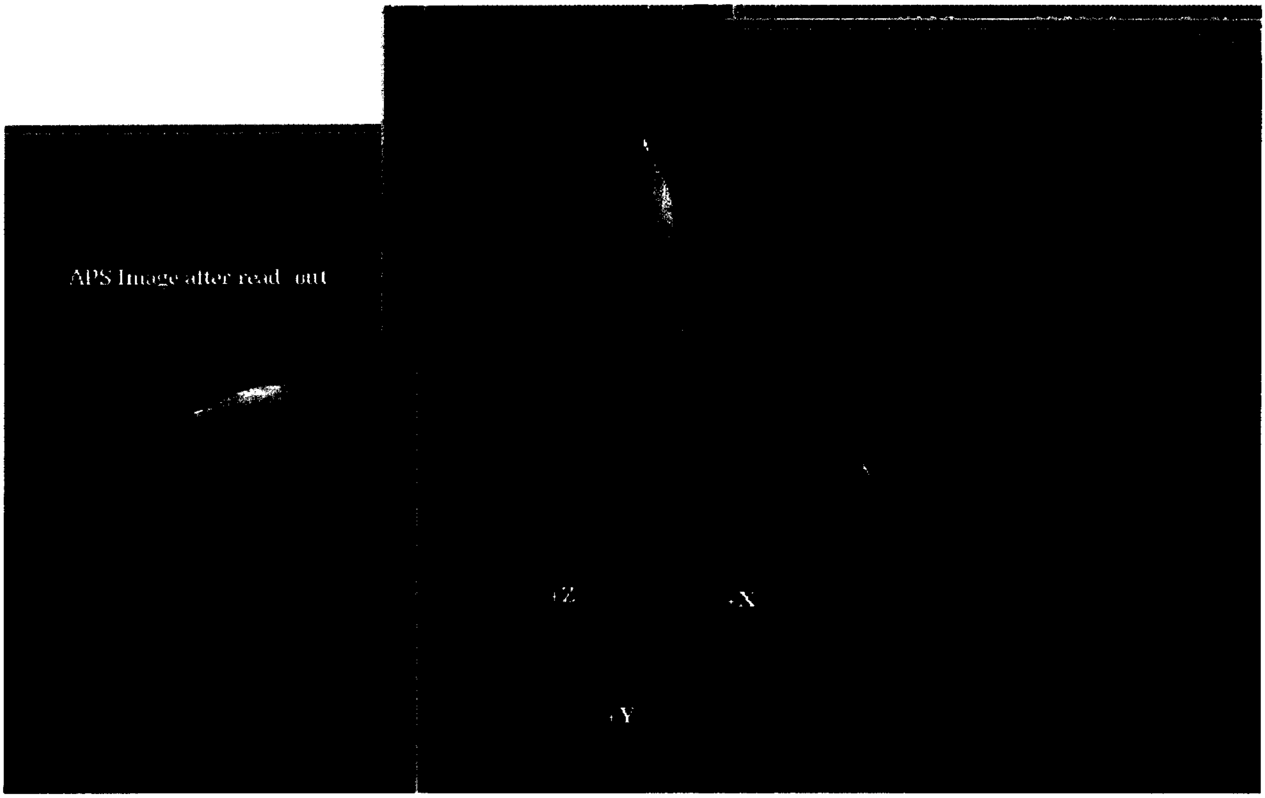


Figure 6 Asteroid Observation Synthesis Example

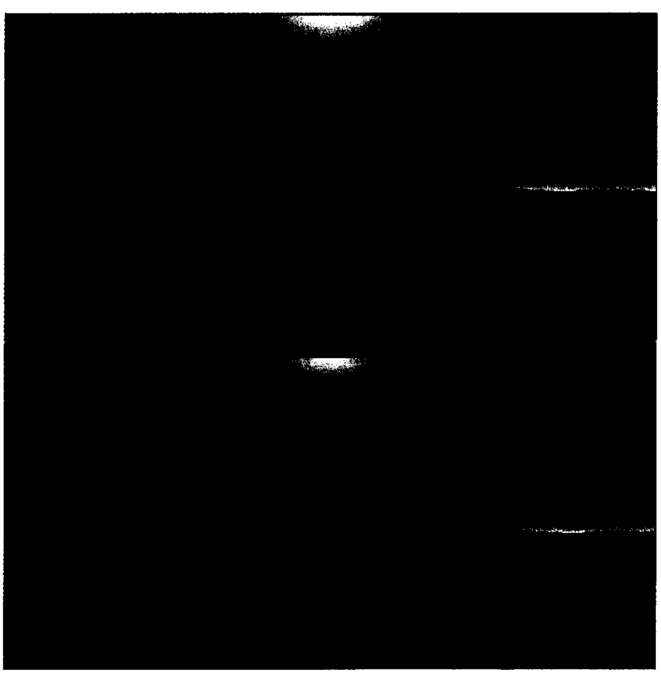


Figure 7 Stereo Image Synthesis Example

The SRMS capability will be made available at the PDC at JPL during 2003 so that the design products can be validated with respect to achieving science-return. In order to support the interactive design validation, two additional technology components will be integrated, distributed mission simulation and 4D broadcasting service. The distributed mission simulation capability will enable parallel execution of large number of subsystem properties, rapid parameter space exploration, and interactive high-fidelity science data product generation.

ACKNOWLEDGEMENTS

The work described in this paper was performed at Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] M. Lee, R. Weidner, "Science Observation and Mission Lifecycle," SpaceOps-2002, Houston, Texas, October 9-12, 2002.

[2] U. Eco., "A Theory of Semiotics," Indiana University Press, 1976.

[3] "Languages," Blackwell Scientific Publications, 1987.

[4] M. Devitt, K. Sterelny, "Language & Reality," A Bradford Book, The MIT Press, Cambridge, Massachusetts, 1995.

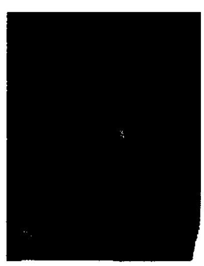
[5] M. Lee, R. Weidner, W. Lu, "Mission Lifecycle Modeling and Simulation," 2000 IEEE Aerospace Conference, Big Sky, Montana, March 2000.

[6] R. J. Weidner, "A Component-based Implementation of Agents and Brokers for Design Coordination," submitted to 2001 IEEE Aerospace Conference, Big Sky, Montana, March 2001.

[7] M. Lee, R. Weidner, W. Lu, "Design-based Mission Operation," 2001 IEEE Aerospace Conference, Big Sky, Montana, March 2001.

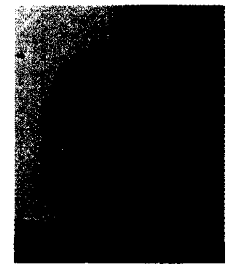
[8] M. Lee, R. Swartz, and R. Weidner, "*SceneGen,*" *NASA* Tech Brief, 1997.

[9] M. Lee, R. Weidner, "In-situ Site Knowledge System", 2002 IEEE Aerospace Conference, Big Sky, Montana, March 2002.

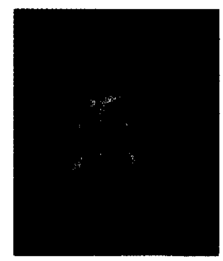


Meemong Lee *is a principal technologist at the Jet Propulsion Laboratory. She leads mission system modeling and simulation research activities in support of the Project Design Center and advanced mission studies. Her current development activities include Science-return validation system and Autonomous In-situ Science Testbed. She has a bachelor's degree in Electronics Engineering from Sogang University in S. Korea, a master's degree in Computer Science and a doctoral degree in Electrical Engineering from Oklahoma State University.*

Richard J. Weidner *is a principal technologist and a technical group supervisor of the MisOsion Simulation and Instrument Modeling Group at JPL. He supported automated mosaic sequence generation and real-time Imp camera image interpretation during the Mars Pathfinder. He invented Micro-Helm for real-time telemetry visualization in support of the Mars Odyssey. His current research activities include intelligent mission model agents, a 4-D broadcasting system, and Micro-Helm/Ripples. He has a bachelor's degree, a master's degree, and doctoral degree in Electrical Engineering from Oklahoma State University.*



Shin-Ywan Wang *is a senior engineer of the Mission Simulation and Instrument Modeling group at JPL. She supports virtual mission simulation system for integrating the subsystem performance analysis tools, designing mission scenario language, and developing instrument property modeling methods. She has a bachelor's degree in Electronics Engineering from National Chiao-Tung University, Taiwan, and a master's degree and doctoral degree in Electrical and Computer Engineering from University of California, Irvine.*



Brian Lininger *is a staff member of the Mission Simulation and Instrument Modeling Group at JPL. He develops a software framework for streamlining the interface between the Project Design Center at JPL and the Mission Simulation and Instrument Modeling group. His recent work includes science-return centric design parameter database composition, automated Excel-workbook generation for subsystem property modeling, and a telemetry visualization coordinator for Micro-Helm. He has a bachelor's degree in Computer Science from Cal Poly Pomona.*